



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/597,524	06/20/2000	Harry J. Beatty, III	FIS9-1999-0319-US1	5261
29505	7590	01/14/2004	EXAMINER	
DELIO & PETERSON, LLC 121 WHITNEY AVENUE NEW HAVEN, CT 06510			ALI, SYED J	
			ART UNIT	PAPER NUMBER
			2127	
DATE MAILED: 01/14/2004				

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/597,524

Applicant(s)

BEATTY, III ET AL.

Examiner

Syed J Ali

Art Unit

2127

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 31 October 2003.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-16 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-16 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. §§ 119 and 120

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
* See the attached detailed Office action for a list of the certified copies not received.
- 13) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.
a) ☐ The translation of the foreign language provisional application has been received.
- 14) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

Attachment(s)

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____
- 4) ☐ Interview Summary (PTO-413) Paper No(s). _____
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: _____

DETAILED ACTION

1. This office action is in response to Amendment A, paper number 5, which was filed October 31, 2003. Claims 1-16 are presented for examination.

2. The text of those sections of Title 35, U.S. code not included in this office action can be found in a prior office action.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. Claims 1-5, 7-9, and 11-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Achenson et al. (USPN 6,477,586) (hereinafter Achenson) in view of LiVecchi (USPN 6,427,161) (hereinafter LiVecchi).

As per claim 1, Achenson discloses a method of parallel processing comprising:

providing a first thread which represents an independent flow of control managed by a program structure (col. 5 line 26 - col. 6 line 63, "The thread and queue pairs, in blocks 40, 42, 44 as shown in Process 1 are referred to as worker threads in the example of the preferred

Art Unit: 2127

embodiment. Such threads are involved in the parallel processing of RPC request and response messages”);

providing a second thread which represents an independent flow of control managed by a program structure separate from the first thread (col. 5 line 26 - col. 6 line 63, “Within process 2A is illustrated a dispatcher thread and associated queue, shown in block 100”);

using the second thread to prepare work for the first thread to process (col. 5 line 26 - col. 6 line 63, “The dispatcher thread, as shown in blocks 100 and 102, in each process is responsible for passing an RPC request message to the appropriate available thread from the pool of worker threads within the process to permit the RPC message to be processed”)

placing the work prepared by the second thread in queue for processing by the first thread (col. 5 line 26 - col. 6 line 63, “The dispatcher thread shown in block 100 determines which of the worker threads found in process 2A of block 32, is appropriate to handle the RPC request message. The message is placed in the appropriate queue relating to the thread which can handle the request”);

LiVecchi discloses the following limitations not shown by Achenson, specifically said first thread having two states, a first state processing work for the program structure (col. 3 lines 15-67, “If a request is waiting, the thread removes the request from the queue, and begins to process it”) and a second state undispached awaiting work to process (col. 3 lines 15-67, “The idle thread may then ‘sleep’, whereby a system timer is used to cause the thread to wait for a predetermined period of time, and then ‘awaken’ to recheck the queue to see if work has arrived”);

Art Unit: 2127

if the first thread is awaiting work to process when the work prepared by the second thread is placed in the queue, dispatching the first thread and using it to process the work in the queue (col. 3 lines 15-67, "If a request is waiting, the thread removes the request from the queue, and begins to process it");

if the first thread is processing other work when the work prepared by the second thread is placed in the queue, using the first thread to complete processing of the other work, access the work in the queue, and then process the work in the queue (col. 3 lines 15-67, "As each thread completes the work request it has been processing, it looks on the queue for its next request").

It would have been obvious to one of ordinary skill in the art to combine Achenson with LiVecchi because Achenson fails to specifically disclose how to handle various states of execution for threads. LiVecchi thus provides a mechanism for handling the various states of execution, including threads that are active and currently processing tasks, idle and waiting for work, and active and waiting for work. These mechanisms for handling different states of execution are well known in the art, and although not specifically disclosed by Achenson, would apply to the multithreading architecture of Achenson very easily.

As per claim 2, Achenson discloses the method of claim 1 wherein the second thread continues to place additional work in the queue, and the first thread sequentially processes the additional work in the queue as it completes processing prior work (col. 5 line 26 - col. 6 line 63, "The dispatcher thread shown in block 100 determines which of the worker threads found in process 2A of block 32, is appropriate to handle the RPC request message. The message is placed in the appropriate queue relating to the thread which can handle the request").

As per claim 3, Achenson discloses the method of claim 1 wherein the second thread marks the work placed in the first thread queue as not complete (col. 5 line 26 - col. 6 line 63, “When the RPC request message is forwarded via connection 90, the hConn value of the RPC request is set to null”, wherein the system knows that it is a new request if the hConn value is null).

As per claim 4, Achenson discloses the method of claim 1 wherein if the first thread is processing other work when the work prepared by the second thread is placed in the queue, and when the first thread completes processing of the work in the queue, using the first thread to mark the completed work as complete, wherein subsequent work from the second thread is made to wait until the previous work in the first thread is marked complete (col. 5 line 26 - col. 6 line 63, “The message receiver thread of block 72, upon receiving the message, updates the hConn value to the connect id of connection 90”, “the RPC response is returned to Process 1 of block 30 via connection 90. The correct connection is determined by reference to the hConn value found in the RPC message”, wherein the determination for what path to use in sending the response is made from the hConn value in the RPC message, and also the updating of that value indicates that the message has been serviced).

As per claim 5, LiVecchi discloses the method of claim 1 wherein the first thread is reused to process other work (col. 3 lines 15-67, “As each thread completes the work request it has been processing, it looks on the queue for its next request”).

As per claim 7, it is rejected for similar reasons as discussed above for claims 1-3. Specifically, all the limitations of the present claim are the same as if claims 1-3 had been presented as a single claim. Therefore, the citations above for claims 1-3 show where in the modified Achenson the present limitations can be found. Further, the motivation for combining Achenson with LiVecchi is discussed above in reference to the discussion of claim 1.

As per claims 8-9, they are rejected for similar reasons as claims 4-5, respectively.

As per claims 11-15, they are rejected for similar reasons as discussed above for claims 1-5, respectively. Specifically, both Achenson and LiVecchi are disclosed in relation to computer systems, with program storage devices tangibly embodying a program of instructions executable by a machine to perform the method steps of claims 1-5 (see Figs. 1-2 of LiVecchi). Further, all of the limitations of claims 11-15 are disclosed by the combination of Achenson and LiVecchi, as shown by the citations in the discussion of claims 1-5. Additionally, the motivation for combining Achenson and LiVecchi is discussed above in reference to the discussion of claim 1.

5. Claims 6, 10, and 16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Achenson in view of LiVecchi in view of Voll et al. (USPN 6,170,018) (hereinafter Voll).

As per claim 6, Voll discloses the following limitations not shown by the modified Achenson, specifically the method of claim 4 wherein the program structure destroys the first thread after it completes a desired amount of work (col. 23 lines 28-37, "When processing by a particular thread completes, e.g., by execution of the computer instruction 'return', the thread is destroyed").

It would have been obvious to one of ordinary skill in the art to combine the modified Achenson with Voll since once a thread has completed processing the work that is allotted to it, or completed a sufficient amount of work, it would be to the benefit of the system to destroy that thread. The reason for this is that an inactive thread occupies memory space that could be used for other purposes. By freeing these resources, more resources can be allocated to other threads, thereby increasing the throughput of the entire system.

As per claims 10 and 16, they are rejected for similar reasons as claim 6.

Response to Arguments

6. Applicant's arguments filed October 31, 2003 have been fully considered but they are not persuasive.

7. Applicant argues on page 8, "*The hypothetical combinations of Achenson and LiVecchi as cited does not present a prima facie case of obviousness [since]...Achenson discloses nothing about a thread having two states, and using a second thread to prepare work and place it in a queue for processing by the first thread when it is completed processing any other work.*"

Art Unit: 2127

Applicant adds that the addition of LiVecchi fails to account for this alleged deficiency of Achenson since in LiVecchi, *"the worker threads are themselves responsible for checking a passive socket queue to determine if there are any connection requests, and if a request is waiting the thread removes the request from the queue and begins to process it. If no request is waiting, the thread then becomes an idle thread which then sleeps. Thus, in this...system cited by the Examiner where the worker threads have two states, it is specifically disclosed that no dispatcher thread is utilized."*

While it was noted that Achenson does in fact fail to specifically disclose a worker thread having two states, Achenson does in fact disclose using a second thread to prepare work and place it in a queue for processing by the first thread. This is achieved through the use of a dispatcher thread within each process that receives a request and forwards the request along to a thread that is capable of servicing that request. However, since Achenson fails to specifically disclose multiple states of operation, LiVecchi is cited to show a method of operating a thread that is in accordance with the disclosure of Achenson, while allowing a thread to remain in an idle state or active state, depending on the current processing load. Although LiVecchi does not use a dispatching thread for preparing work, this aspect of the claimed invention comes from Achenson rather than LiVecchi. LiVecchi is relied upon only to show how the worker thread acts when work is not present in the queue, or when the thread is currently processing work and an additional request is placed in the queue. The method by which work actually arrives at the queue is cited from Achenson, wherein a dispatching thread receives a request, determines the worker thread that is best suited to service the request, and places the request in the processing queue associated with that thread.

8. Applicant argues that the rejection of claims 2-4, 7-8, and 12-14 is deficient since the combination of Achenson and LiVecchi fails to disclose a dispatcher thread. However, this argument has been addressed above in item 7. Thus, the rejection of claims 2-4, 7-8, and 12-14 stands as presented in the previous Office action, and reiterated above.

9. Applicant argues on page 10, *“one of ordinary skill in the would not look to this portion of LiVecchi in connection with applicant’s claimed process using a second thread to provide work in a queue to a first, two-state thread”* since *“this aspect of the various processes disclosed therein is from the embodiment that is ‘implemented without using a dispatcher thread.’”*

It is noted that this citation does indeed come from the embodiment that is implemented without a dispatcher thread. However, LiVecchi is cited in an effort to show how the worker thread removes work to be processed from its processing queue, not how the work arrives at the queue. For how the work arrives at the queue, Achenson is cited, which discloses a dispatching thread receiving a request, determining the worker thread that is best suited to service the request, and placing the request in the processing queue associated with that thread.

10. Applicant argues that the rejection of claims 6, 10, and 16 is deficient since the combination of Achenson, LiVecchi, and Voll fails to disclose a dispatcher thread. However, this argument has been addressed above in item 7. Thus, the rejection of claims 6, 10, and 16 stands as presented in the previous Office action, and reiterated above.

Conclusion

11. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

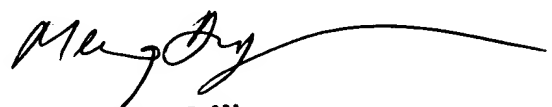
Any inquiry concerning this communication or earlier communications from the examiner should be directed to Syed J Ali whose telephone number is (703) 305-8106. The examiner can normally be reached on Mon-Fri 8-5:30, 2nd Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai T An can be reached on (703) 305-9678. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.



Syed Ali
January 5, 2004



MENG-AI T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100